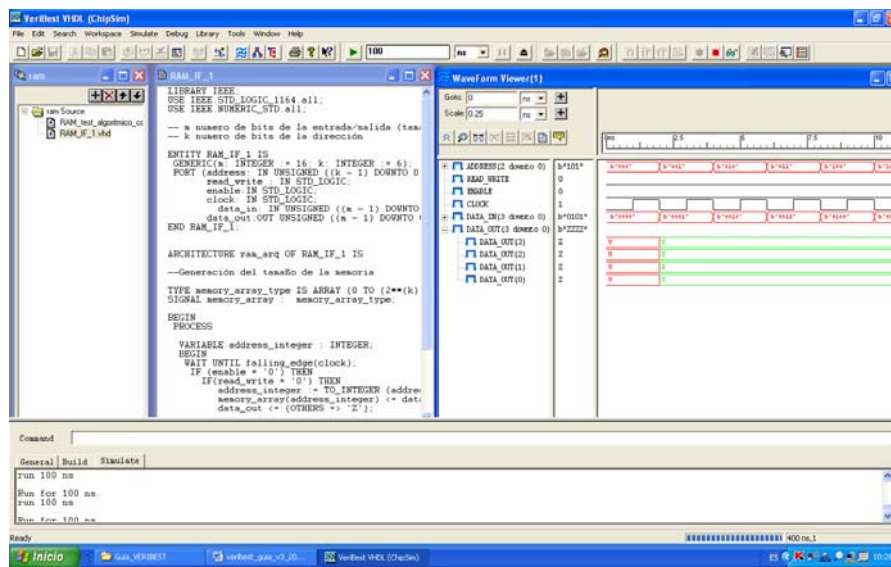



SISTEMAS DIGITALES

Guía de utilización del simulador Veribest




Veribest es un simulador de modelos de circuitos digitales descritos en VHDL.

El objetivo de esta guía es facilitar un primer contacto con el uso del simulador. No se trata de una guía detallada. La utilización de los comandos y funciones más comunes de la herramienta, se presentan a través de la realización de un ejemplo sencillo, en el cual aborda la edición de un fichero, la compilación y la visualización de las formas de onda de los resultados de simulación.

La herramienta tiene una **ayuda on line**, que se recomienda utilizar cuando se considere necesaria. Para activarla, hay que seleccionar el botón de la ayuda **on line** (véase Figura-1 ), aparecerá un símbolo de interrogación que se puede desplazar por la ventana, y que deberá posicionarse sobre el icono, ventana, palabra etc., que sea motivo de consulta, apareciendo a continuación la correspondiente información de ese aspecto específico. Asimismo, puede observarse en la barra de herramientas la función **Help** que contiene un menú desplegable con información sobre el uso de la herramienta.

1) Arranque del simulador

La herramienta funciona bajo el sistema operativo Windows de Microsoft. Para arrancar la herramienta se ha de seguir la ruta que se indica a continuación o bien crearse un icono en el escritorio , y acceder a la herramienta a través de él:

Todos los programas → VeriBest VB99.0 → VeriBest VHDL simulator → VeriBest VHDL

La ventana del simulador con indicación de algunas de las funciones de la barra de herramientas y del espacio de trabajo se muestra en la siguiente figura:

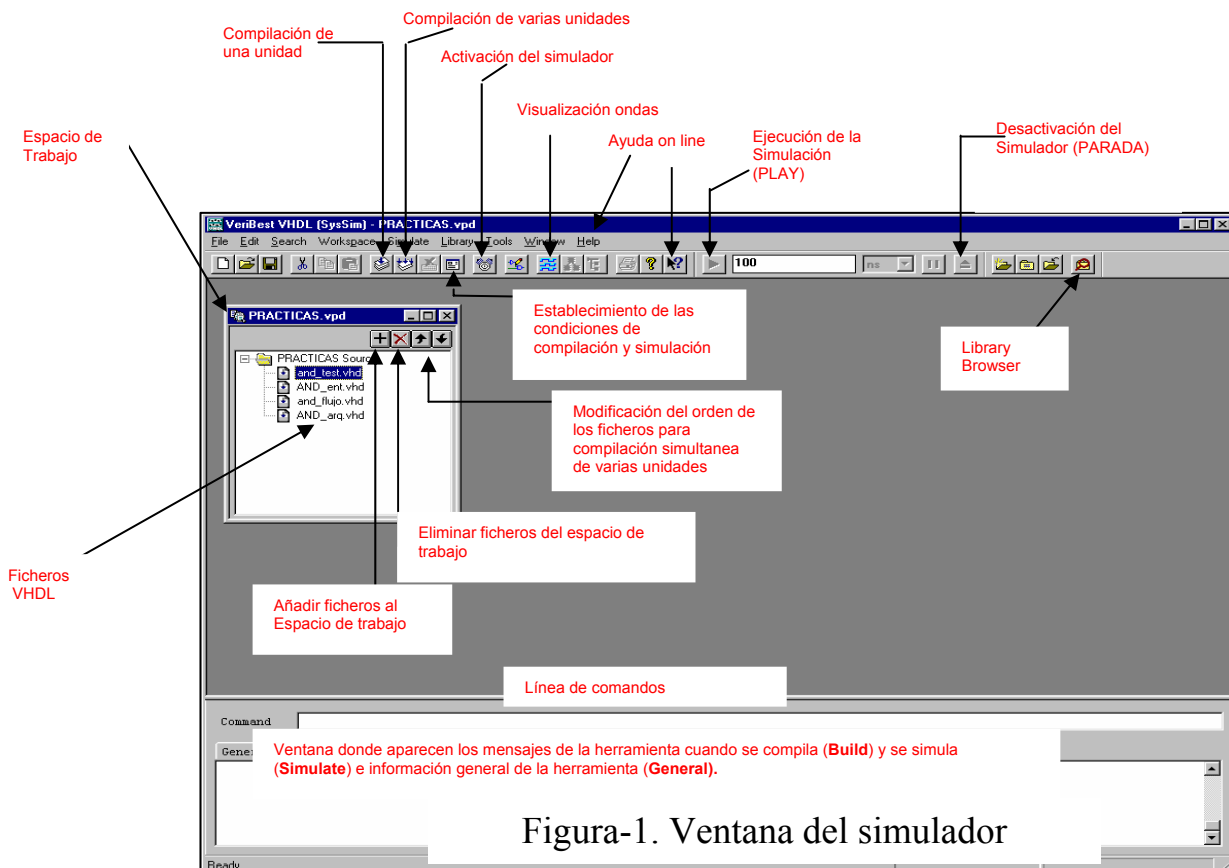
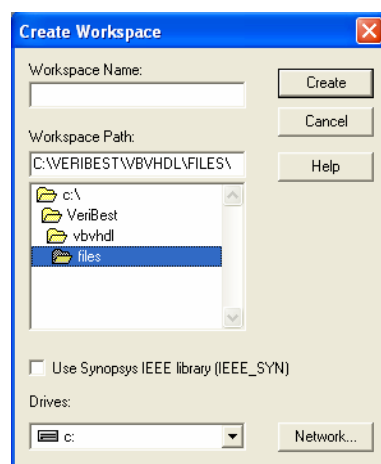


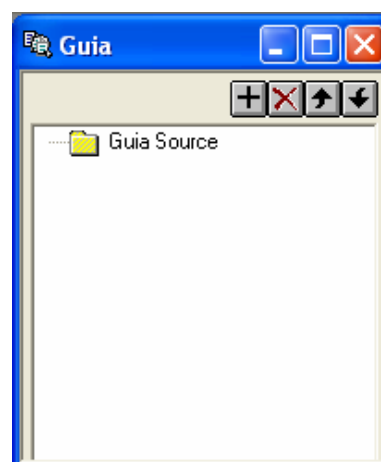
Figura-1. Ventana del simulador

2) Creación de un espacio de trabajo

- Cuando se comienza a trabajar por primera vez con la herramienta, no hay ficheros VHDL dentro de la misma, solamente las bibliotecas propias del simulador. Entonces lo primero que hay que hacer es crearse un espacio de trabajo, seleccionando en la barra de herramientas **Workspace** y dentro de ella **New**.
- Nos aparecerá una ventana pidiéndonos el nombre del espacio de trabajo, que puede ser cualquiera, por ejemplo **GUIA**, y ruta de la carpeta donde se va a guardar. La herramienta por defecto sugiere utilizar la carpeta **files**, aunque se permite utilizar cualquier otra que decida el usuario (por ejemplo **PRACTICAS_VHDL**), a continuación activar el botón **Create**.



- De forma inmediata aparece otra ventana con el nombre del espacio de trabajo que hemos creado (**GUIA**). Dentro de ella, aparece una serie de botones cuya utilidad comentaremos más adelante o a lo largo del desarrollo de las sesiones de prácticas, y una carpeta con el nombre **GUIA source**, que es donde se van a depositar todos los ficheros que escribamos dentro de este espacio de trabajo.

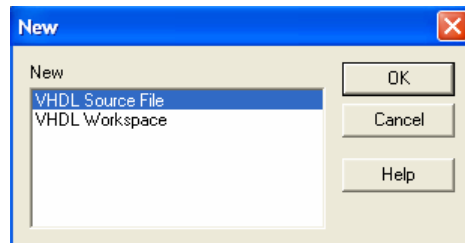


MUY IMPORTANTE, SE PRODUCEN ERRORES SI:

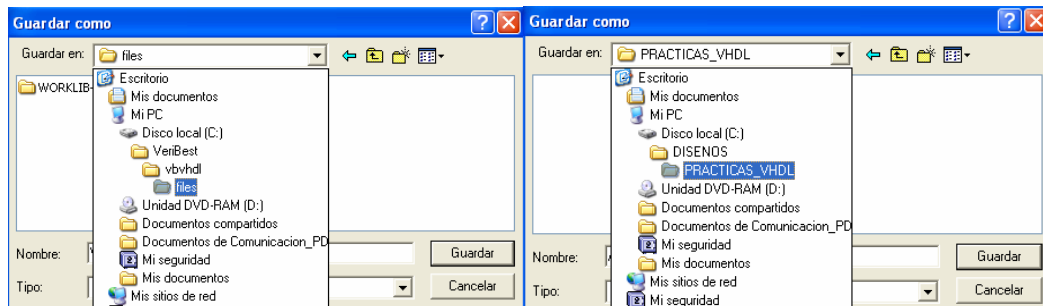
- Los espacios de trabajo, y las **rutas de acceso** a los directorios o carpetas donde van dejarse los diseños contienen: **espacios en blanco, eñes, y/o acentos**.
- Los **nombres de los ficheros** contienen **acentos**.

3) Edición de un fichero nuevo

- Para editar un nuevo fichero se selecciona en la barra de herramientas **File (New)**, a continuación aparece una nueva ventana que nos da las opciones de **VHDL source file o VHDL Workspace**, elegimos la primera opción, ya que lo que queremos es editar un fichero nuevo y no crear otro espacio de trabajo, ya lo tenemos creado en el paso anterior (2), aunque también se puede crear desde aquí, si se desea.



- Nos aparece una ventana con un nombre por defecto (**VHDL1**), que nosotros vamos a personalizar salvándolo con el nombre del fichero que nos interese. El ejemplo que vamos a escribir es una entidad de una puerta **and** de dos entradas, por tanto, un posible nombre para ese fichero podría ser el tipo de la puerta, y su número de entradas, y la extensión **ent** para indicar que contiene solamente una entidad (*véase la nomenclatura aconsejada para denominar a los ficheros que contienen las unidades de diseño, incluida al final de este documento*). El nombre del fichero que contiene la entidad, que es lo que estamos guardando ahora, y el nombre de la entidad pueden ser iguales o no, aunque se aconseja que sean los mismos (**Save as AND2_ent.vhd**). El lugar donde se guarde puede ser dentro de la carpeta **files** de la herramienta o bien en la que nosotros **hayamos decidido** (**PRACTICAS_VHDL**), *se aconseja hacerlo en el mismo lugar donde se ha creado el espacio de trabajo*. Se observará que tras esta acción, aparece en la ventana el nombre del fichero.



- A partir de este momento ya podemos introducir nuestro código. Obsérvese en el ejemplo que se indica a continuación, que las palabras reservadas del lenguaje aparecen escritas con mayúsculas y las demás con minúsculas, y aunque el compilador no distingue entre mayúsculas y minúsculas, se recomienda esta buena práctica, ya que visualmente se interpreta mejor el código. Otros aspectos que siempre han de incluirse en el fichero para documentar y actualizar el diseño, es una cabecera en la que se indique: el propietario del mismo y los derechos de copia, el proyecto al cual pertenece el diseño, el nombre del diseño, el nombre del fichero que lo contiene, el autor, la fecha, la versión, un resumen del mismo, las modificaciones y otra información que se considere relevante para documentar mejor el trabajo.


```

-----
-- © Universidad Politécnica de Madrid
-- Se permite copia para fines de estudio
-----
--
-- Proyecto           : Guia Veribest
-- Diseño            : Puerta and de dos entradas
-- Nombre del fichero : and2_ent.vhd
-- Autor             : V. Rodellar
-- Fecha             : 28/1/2010
-- Versión           : 3.0
-- Resumen           : Este fichero contiene la entidad de una puerta
--                     AND de dos entradas
--
--
-- Modificaciones:
--
-- Fecha      Autor      Versión      Descripción del cambio
-----
--12-10-05   VRB         2.0          Inclusión de cabecera
--28-1-10    VRB         3.0          Modificación a datos tipo
--                                     STD_LOGIC
-----

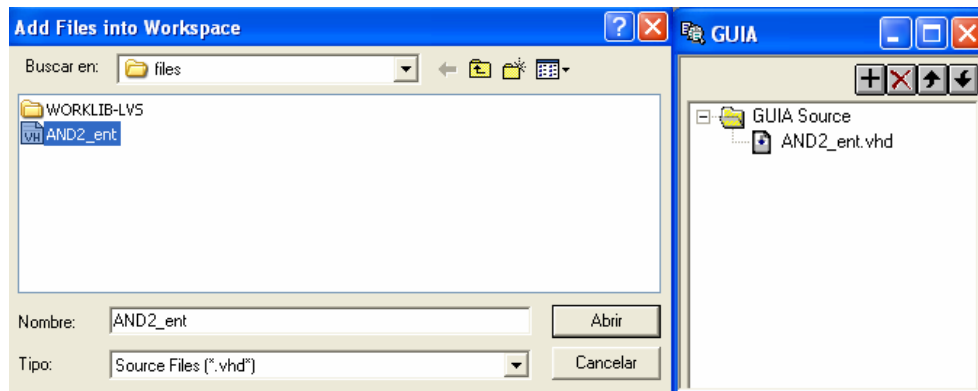
LIBRARY ieee;
USE ieee.STD_LOGIC_1164.all;

ENTITY and2 IS
  PORT (
    e0,e1      : IN STD_LOGIC;
    s          : OUT STD_LOGIC
  );
END and2;
-----


```

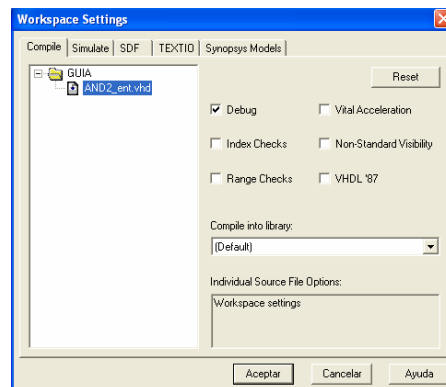
A continuación almacenamos el diseño (**Save**). Se debe observar que la ventana del espacio de trabajo permanece abierta (**GUIA** o bien **PRACTICAS_VHDL Source**, según lo hayamos denominado), y lo primero que vamos a hacer es añadir el fichero que acabamos de escribir a nuestro espacio de trabajo. Para ello, activaremos la tecla **+** del espacio de trabajo (véase Figura-1 ) , nos saldrá una nueva ventana, **Add Files into Workspace**, y sobre ella, seleccionamos el fichero **AND2_ent.vhd**. Se observará entonces, que el fichero

se añade a la carpeta **GUIA Source**. A partir de este momento se puede cerrar la ventana donde hemos editado el fichero.



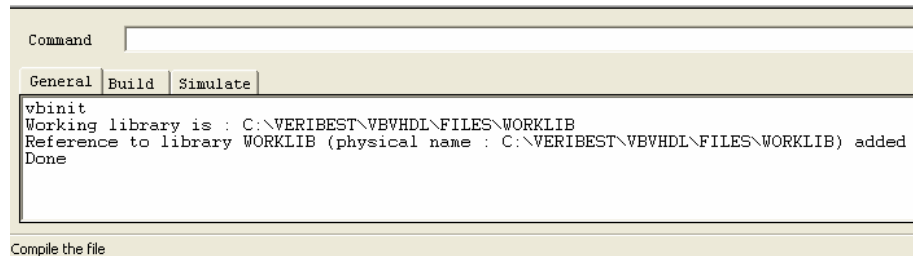
4) *Compilación*


- Lo primero que hay que establecer son las condiciones de compilación. Para ello hay que ir de nuevo a la barra de herramientas, seleccionar **Workspace** y dentro de ella **Settings** o bien activar el correspondiente botón (véase Figura-1 ). Aparecerá una nueva ventana con la carpeta **GUIA Source**, y si damos dos clics sobre ella se abrirá, apareciendo el fichero **AND2_ent.vhd**. Seleccionamos el fichero anterior, y de las posibles opciones que nos da la ventana seleccionamos **Compile y Debug**.





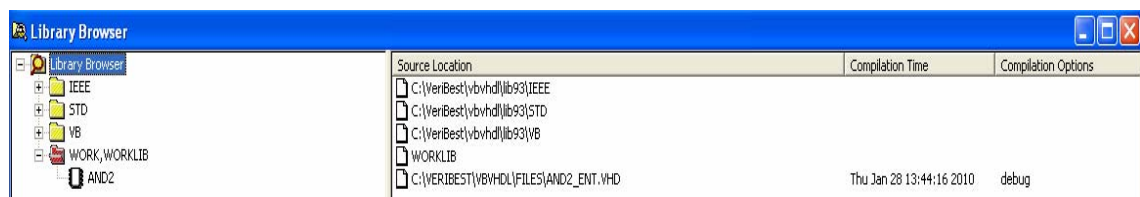
- Una vez establecidas las condiciones de compilación, esta se puede lanzar de dos formas:
- Desplegando **Workspace** en la barra de herramientas, aparece la opción **Compile AND2_ent.vhd**. Cuando esta opción se activa en la ventana inferior del simulador (**General, Build, Simulate**) (véase la Figura-1), aparece una serie de mensajes relativos a las distintas acciones que va realizando la herramienta. Si hemos seleccionado **General** nos informa de todas las acciones que se van realizando, si se selecciona **Build** nos dará información

sólo de cómo se ha realizado la compilación, los errores de sintaxis que se hayan podido cometer, y el número de sentencia donde está, y por último **Simulate**, nos proporciona únicamente información de la fase relativa a la simulación. En el caso que acabamos de describir, la información obtenida se reproduce a continuación:



- Seleccionando el fichero **AND2_ent.vhd** en la ventana del espacio de trabajo y activando en la barra de herramientas el botón para compilación de una sola unidad o de todas las que hay en el espacio de trabajo, en cuyo caso habrá que ordenarlas de menor a mayor en la jerarquía (véase Figura-1 ). Si se hace esto, se observará los mismos efectos en la ventana inferior del simulador que en el caso anterior.

Una vez que la compilación no da errores, se almacena el fichero que se ha compilado (**SAVE o** ). La unidad compilada se almacenará en la librería **WORK**. Para ver donde está ubicada esta unidad de diseño, en la barra de herramientas se selecciona **Library** y dentro de ella **Library Browser** o bien se activa el correspondiente botón (véase Figura-1 ), aparecerá una nueva ventana con unas carpetas resaltadas en amarillo (IEEE, STD, etc.), estos son los nombres lógicos de las librerías que están accesibles a la herramienta. La carpeta **VB** (VeriBest) esta a disposición de diseñador para que allí deposite sus diseños, y cumple la misma misión que el directorio **PRACTICAS_VHD** como hemos mencionado en el punto 2 (creación del espacio de trabajo). También aparece una carpeta resaltada en rojo con el nombre de **WORK, WORKLIB**, si se abre esta librería aparece el símbolo de un chip con el cuerpo en blanco que lleva asociado el nombre de nuestra entidad AND2, **¡no el nombre del fichero donde se encuentra almacenada!** La parte derecha de esta ventana nos da la información de la ruta o nombre físico de las librerías y ficheros, la fecha de cuando se compilaron y las opciones de compilación que se utilizaron. En la parte inferior de la ventana, en el caso de haber seleccionado **General**, aparecerá la palabra **libbrows**, que no es más que la abreviatura de **Lib**rary **Brows**er.



5) Preparación de un diseño

- **Descripción arquitectura**

Todavía no podemos simular ningún diseño, ya que para ello es necesario que la entidad tenga asociada al menos una arquitectura y un fichero de test. Vamos entonces a escribir una arquitectura en estilo flujo de datos para la entidad de la puerta **and** anterior, utilizando simplemente el operador AND, y siguiendo los pasos indicados para la edición en el punto **3**). Para tener una mayor flexibilidad en la configuración de entidades con distintas arquitecturas, en este ejemplo hemos realizado la entidad y arquitectura en ficheros separados, aunque las dos unidades de diseño pueden compartir el mismo fichero. Esta última posibilidad es aconsejable en el caso de que solo se vaya a tener una sola arquitectura asociada a una entidad, ya que ello reduce el número de ficheros. El nombre que le daremos al fichero será **AND2_flujo_arch.vhd** (*véase la nomenclatura aconsejada para denominar a los ficheros que contienen las unidades de diseño, incluida al final de este documento*), y a la arquitectura **flujo**.

```
-----
-- © Universidad Politécnica de Madrid
-- Se permite copia para fines de estudio
-----
-- Proyecto           : Guia Veribest
-- Diseño             : Arquitectura para una puerta AND de dos
--                   : entradas
-- Nombre del fichero  : and2_flujo_arch.vhd
-- Autor              : V. Rodellar
-- Fecha              : 30/5/2005
-- Versión             : 1.0
-- Resumen            : Este fichero contiene una arquitectura en
--                   : estilo flujo
--
--
-- Modificaciones:
--
-- Fecha      Autor      Versión      Descripción del cambio
-----
-- 11-10-05   VRB        1.0          Inclusión de cabecera
-----

ARCHITECTURE flujo OF and2 IS
-- Parte declarativa

BEGIN
-- Descripción de la arquitectura

    s<= e0 AND e1;

END flujo
-----
```

El fragmento de código anterior tiene un error de sintaxis, en la última sentencia falta el identificador de terminación de línea (;). Cuando compilemos la

arquitectura (siguiendo los pasos indicados en el punto 4) en la ventana de mensajes, aparecerá el siguiente:

```

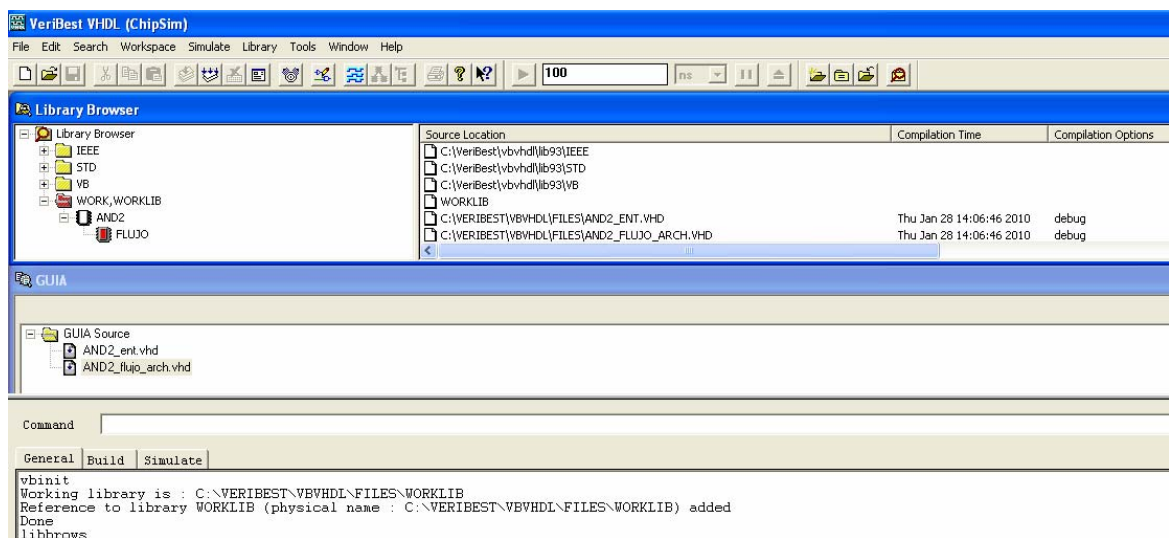
Command
General Build Simulate
vc comp C:\VeriBest\vbvhdl\files\AND2_flujo_arch.vhd
VeriBest VHDL Compiler - 15.00.00.25

Compiling Architecture FLUJO of AND2

-----
37: ^
[Error] Unexpected End of File
Done

```

A continuación se corregirá el error de sintaxis, se volverá a recompilar el diseño y a añadir en el espacio de trabajo. En este momento, se aconseja explorar la librería **WORK, WORKLIB**, como se indica en el punto 4). Se observará, que colgada de la entidad **AND2** aparece el símbolo de un chip con el cuerpo en rojo que tiene el nombre de nuestra arquitectura **FLUJO**, todas las arquitecturas que asociemos a esta entidad ocuparan el mismo lugar en la jerarquía.



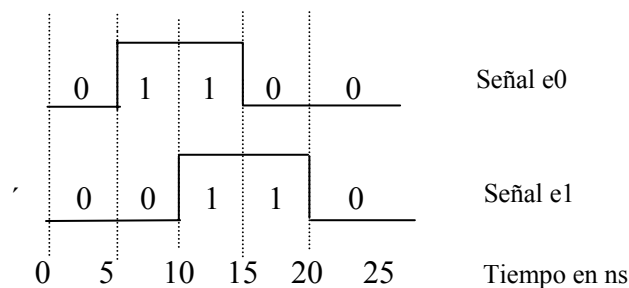
- **Descripción del test**

- **Preparación de los estímulos de entrada**

El test, siempre que las dimensiones del problema lo permitan, debe contener todas las posibles combinaciones de los valores de las entradas, en nuestro caso, las posibles combinaciones se reducen a las siguientes:

e1	e2	s
0	0	0
0	1	0
1	0	0
1	1	1

Unas posibles formas de onda (drivers) que recogiesen la totalidad de los casos de la tabla anterior serían:



- **Escritura del fichero de test**

Para testear un componente es necesario escribir una entidad y una arquitectura. Las entidades para test no tienen puertos. Y las arquitecturas son de tipo estructural, incluyendo en su parte declarativa el componente que se va a testear, la arquitectura que se le asocia y la declaración de señales.

```

-----
-- © Universidad Politécnica de Madrid
-- Se permite copia para fines de estudio
-----
-- Proyecto           : Guia Veribest
-- Diseño             : Test para una puerta and de dos entradas
-- Nombre del fichero : and2_tb.vhd
-- Autor              : V. Rodellar
-- Fecha              : 30/5/2005
-- Versión            : 1.0
-- Resumen            : Contiene la entidad y la arquitectura para
--                      test. El test es completo, contiene todas las
--                      posibles combinaciones de las entradas
--
-- Modificaciones:
--
-- Fecha      Autor      Versión      Descripción del cambio
-----
--

```

```

-----
LIBRARY ieee;
USE ieee.STD_LOGIC_1164.all;

ENTITY and2_test IS
END and2_test;

ARCHITECTURE test_flujo OF and2_test IS

-- Parte declarativa

-- Declaración del componente que se va a testear
COMPONENT and2 PORT(
    e0, e1 : IN STD_LOGIC;
    s      : OUT STD_LOGIC
);
END COMPONENT;

-- Configuración de la arquitectura. ¡¡No es obligatorio!!
-- Si no se especifica tomaría por defecto la última arquitectura
-- compilada
FOR C1: and2 USE ENTITY WORK.and2(flujo);

-- Declaración de señales
SIGNAL e0, e1, s: STD_LOGIC;

-- Cuerpo de la arquitectura
BEGIN

-- Intanciación del componente a testear y conexionado de puertos
C1: and2 PORT MAP (
    e0 => e0,
    e1 => e1,
    s  => s
);

-- Drivers de las señales de entrada
e0 <='0','1'AFTER 5 NS, '0'AFTER 15 NS;
e1 <='0','1'AFTER 10 NS,'0'AFTER 20 NS;


END test_flujo;
-----

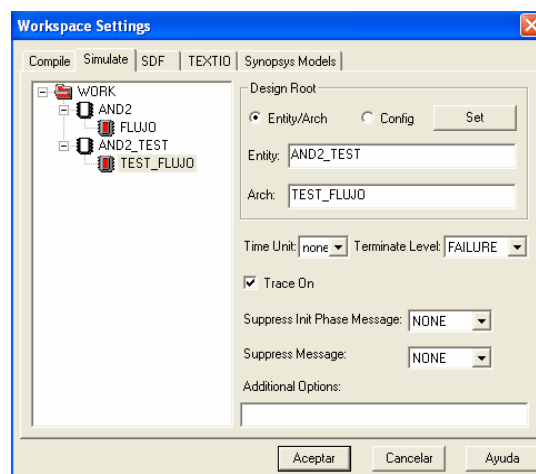
```

Aunque las entidades y arquitecturas pueden ir en ficheros separados, en este caso para simplificar el manejo de ficheros, la entidad para test y su arquitectura se editaran en un mismo fichero, que tendrá por nombre **and2_tb.vhd** (*véase la nomenclatura aconsejada para denominar a los ficheros que contienen las unidades de diseño, incluida al final de este documento*). Se compilará el fichero depurando los posibles errores de sintaxis, y por último se incorporará al espacio de trabajo, siguiendo los pasos mencionados en los apartados anteriores. Se recomienda ver como se han incorporado la nueva entidad y arquitectura a la librería **WORK**.


6) Preparación de la simulación

- **Establecimiento de las condiciones de simulación**

Lo primero que hay que hacer es establecer las condiciones de simulación, para ello se despliega **Workspace** en la barra de herramientas y se selecciona **Settings**, o bien se activa el correspondiente botón (véase Figura-1 ) , como en el caso de la compilación. Pero ahora en la ventana de dialogo que aparece se selecciona **Simulate** y se abre la carpeta **WORK**. Dentro de ella, aparece la entidad **AND_TEST** y su arquitectura asociada **TEST_FLUJO**, así como la entidad **AND2** y su arquitectura **FLUJO**, que es el componente que se desea testear. Hay que simular el **test**, que es donde se ha configurado la entidad **AND2** con la arquitectura **FLUJO**, y se han incluido las correspondientes señales de test. Se selecciona la entidad **AND2_TEST**, se pone el cursor sobre el espacio en blanco que acompaña a la palabra **Entity**, y a continuación se activa el botón de **Set**. Tras esta última acción, se observará que el nombre de la entidad seleccionada aparece copiado en el espacio en blanco. A continuación seleccionaremos la arquitectura **TEST_FLUJO** posicionando el cursor en el espacio en blanco identificado con la palabra **Arch** (abreviatura de arquitectura), y lo activaremos mediante el botón de **Set**, apareciendo el nombre de la arquitectura en el sitio donde habíamos posicionado el cursor. Y por último, activaremos la opción **Trace On** (esta selección nos va a permitir visualizar las formas de onda resultantes de la simulación). La visualización de las formas de onda también puede seleccionarse desplegando la función **Simulate** de la barra de herramientas y activando dentro de ella **Trace**.



- **Activación del simulador**

La activación del simulador se puede hacer por dos procedimientos, desde **Workspace (Execute Simulator)** o bien desde la barra de herramientas activando el símbolo de activación del simulador (véase la Figura-1 ). Se recomienda tener activada la ventana de mensajes en **General**, en ella podremos leer un mensaje similar al siguiente:

```
vbvhdl4ive -d AND2_TEST TEST_FLUJO -tr on -i -dbg
VeriBest VHDL Simulator Version 15.00.00.25.
Starting Simulation ... Thu Jan 28 14:41:24 2010

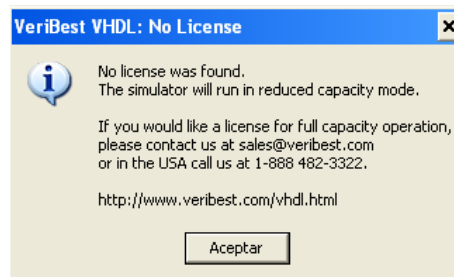
License error: Error reading license file
No such feature exists
Feature:          VBVHDL_CHIP_NT
License path:
C:/Modeltech_6.3c/LICENSE.TXT;C:/Modeltech_6.3c/LICENSE.TXT;
C:/Modeltech_6.3c/LICENSE.TXT
FLEXlm error:    -5,357
License file is C:/Modeltech_6.3c/LICENSE.TXT
Feature is VBVHDL_CHIP_NT.

-----
No license was found.
The simulator will run in reduced capacity mode.

If you would like a license for full capacity operation,
please contact us at sales@veribest.com
or in the USA call us at 1-888 482-3322.

http://www.veribest.com/vhdl.html
```

y la siguiente ventana:



que nos indica esencialmente, que tenemos una licencia de capacidad reducida de uso (hasta 2000 líneas de código), y que no encuentra el fichero de licencias, **no hay que preocuparse**, activar **aceptar** y en ese momento obtendremos el siguiente mensaje:


```
-----
NOTE: Number of component (cell) instances in the design: 1

Checkpointing at simulation time 0 ns.
Checkpoint completed.
Ready to simulate ... Thu Jan 28 14:41:29 2010
```

```
-- Message Summary:  
Total: 0 error(s), 0 warning(s), 0 note(s)
```


Donde se puede observar que no hay errores y que se puede empezar la simulación.

- **Ejecución de la simulación**



En la barra de herramientas (véase Figura-1), aparece el símbolo convencional de **play** , resaltado en verde. Y próximo a él, una ventana que pone 100 y **ns**, esto indica el tiempo de ejecución total de la simulación y las unidades de tiempo asociadas. Ambos valores se pueden cambiar, estos son los valores por defecto. Para ejecutar la simulación hay que activar la tecla del símbolo del **play**. Obteniendo el siguiente mensaje en la ventana inferior:

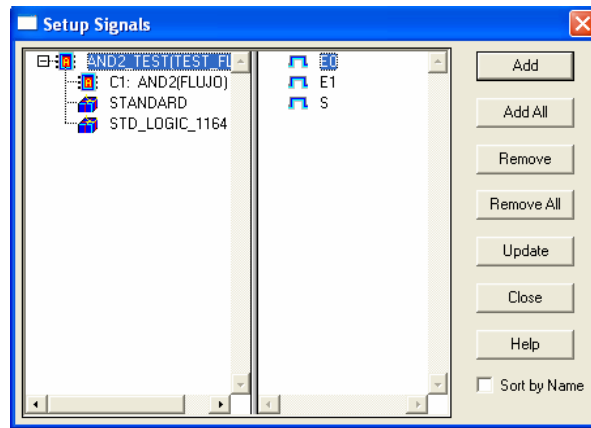
```
run 100 ns  
  
Run for 100 ns.
```

Otra opción para ejecutar la simulación es desplegar **Simulate** en la barra de herramientas y activar **Run**.

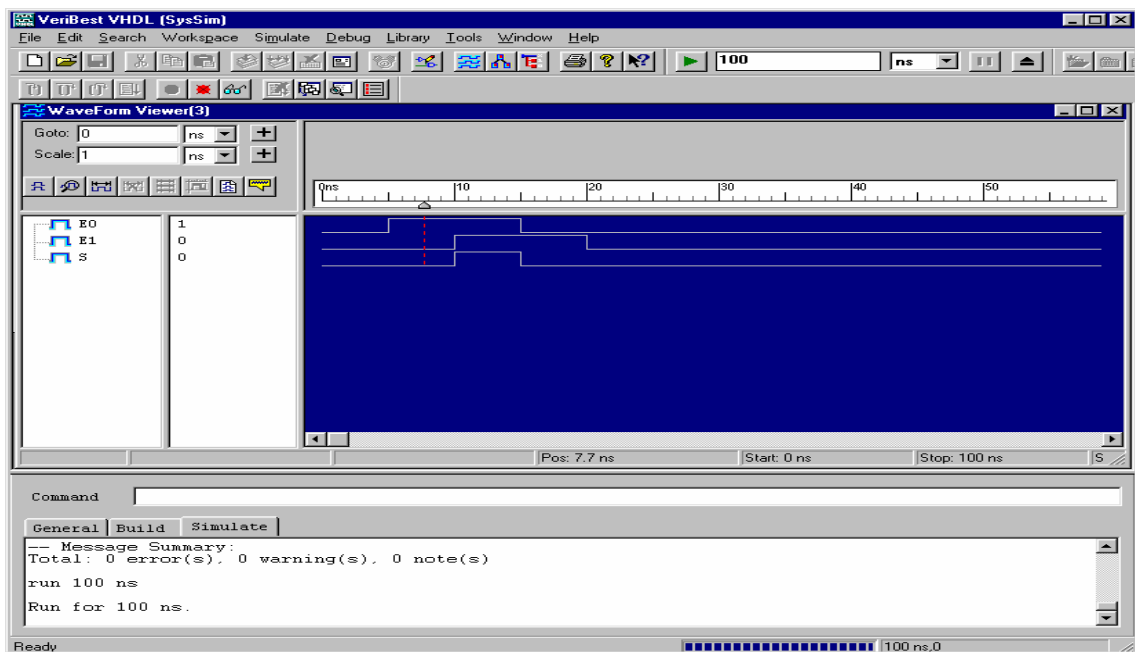
Si se desea desactivar el simulador se selecciona la función **quit** del desplegable **Simulate** o el símbolo de parada en la barra de herramientas (véase Figura-1 ).


7) Visualización de los resultados de simulación

Las formas de onda resultantes de la simulación se pueden analizar activando el botón de visualización de ondas en la barra de herramientas (véase la Figura-1 ). Aparecerá una ventana con el nombre **WaveForm Viewer**, ella contiene una escala de tiempos, una hoja con el fondo azul (cuyo color puede cambiarse mediante **TOOLS -> options -> background**), y un botón con unas formas de onda , activar ese botón. Aparecerá una nueva ventana con el nombre de **Setup Signals**, y dentro de ella, la arquitectura realizada para el test y sus señales asociadas. Podemos entonces seleccionar las señales a visualizar de una en una **Add**, o bien todas de una vez **Add All**.




Observaremos entonces como las señales seleccionadas se incorporan a la ventana **WaveForm Viewer** con sus correspondientes formas de onda, tal como se muestra en la siguiente figura.



La impresión de las formas de onda que aparecen en la pantalla puede realizarse mediante las funciones específicas de la herramienta (, **File-Print**). Aunque en algunas configuraciones de algunas máquinas, las funciones de impresión presentan incompatibilidades con WINDOWS y abortan el simulador. En estos casos, se recomienda capturar la pantalla y pegarla a un fichero de texto.

En la anterior figura, se puede observar la existencia de un cursor, que se puede desplazar por la regleta graduada en unidades de tiempo, y también los nombres de las señales, y los valores lógicos de las mismas en el punto específico donde está

posicionado el cursor. Se recomienda mover el cursor para observar los cambios en los valores de las señales. Así como, experimentar con el zoom y realizar medidas de tiempos entre dos puntos de las señales.

Finalmente si se desea realizar una nueva simulación o cualquier otra acción se debe parar el simulador .

NOMENCLATURA COMUNMENTE USADA PARA DENOMINAR A LOS FICHEROS QUE CONTIENEN A LAS UNIDADES DE DISEÑO

Contenido del fichero	Nombre del fichero
Una entidad y una arquitectura	Nombredelaentidad.vhd
Solo una entidad	Nombredelaentidad_ent.vhd
Solo una arquitectura	Nombredelaentidad_nombrearquitectura_arch.vhd
Más de una arquitectura	Nombredelaentidad_arch.vhd
Solo una configuración	Nombredelaentidad_nombrede laconfiguracion_cfg.vhd
Más de una configuración	Nombredelaentidad_cfg.vhd
Paquete	Nombredelaentidad_pkg.vhd
Test (test bench)	Nombredelaentidad_tb.vhd

GUIA RESUMIDA PARA SIMULAR UN MODELO VHDL CON VERIBEST

- 1) Crear un espacio de trabajo (files o el lugar donde quiera el usuario). La ruta no puede contener eñes, blancos y/o acentos.**
- 2) Editar el fichero con la/s unidades de diseño VHDL (entidad, arquitectura, configuración) y guardarlo (se recomienda en el mismo espacio de trabajo). Los nombres de los ficheros no deben tener acentos).**
- 3) Activar la función debug y compilar la unidad de diseño.**
- 4) Realizar el modelo de test y compilarlo. La entidad está vacía de contenido y el estilo de arquitectura es estructural.**
- 5) Visualizar el contenido de la librería WORK para ver la asociación entre entidades y arquitecturas. Solo contiene las que no tienen errores de compilación.**
- 6) Establecer las condiciones de simulación. Seleccionar la entidad y arquitectura realizada para testear el componente. Activar trace on.**
- 7) Ejecutar la simulación habiendo seleccionado el periodo de tiempo que se desee.**
- 8) Visualizar las formas de onda de los resultados.**
- 9) Parar el simulador**